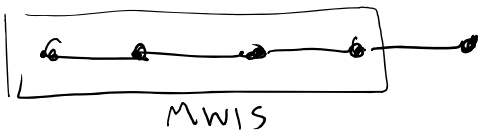
S. KIMMEL

To Create D. P. (dynamic programming) algorithm:

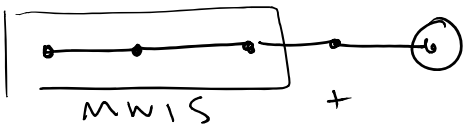
1. Think of form of optimal solution.
   - WMIS on line: $v_n \in S$ (ii) or $n \notin S_n$ (i)

2. How do you write in terms of optimal solution to smaller problem?

(i)



(ii)



3. Create recurence relating value of optimal solution to smaller solutions.

$$A[k] = \max\{A[k-1], A[k-2] + w_k\}$$

(i) (ii)

$A[k]$: weight of MWIS on G_k

4. Store values in an array using for loop

5. Work backwards through array to reconstruct optimal solution

S. KIMMEL

More Dynamic Programming Practice

Sequence Alignment

Problem: How similar are 2 DNA sequences?

Useful for:
- determining closeness of species
- determining when species diverged

DNA changes in specific ways:

GATACA → GATTACA (insertion)

GATACA → GTACA (deletion)

GATACA → GAGACA (mutation)

Ex: How similar are

GATTACA and GAGACA ?

S. KIMMEL

Line up

G A T T A C A
G A - - G A C A

Good: Letters match

Bad: Letter matched with gap

Bad: Letters mismatch

If mostly match, DNA strings are similar!

Sequence Alignment

Input: 2 strings over $\{A, C, T, G\}$ of length n, m $(x_1, \dots x_n)$ $(y_1, \dots y_m)$ (x, y)

Output: Alignment that minimizes penalties

P_{gap} = penalty each time letter matched w/ gap

P_{mm} = " mismatched

Alignment is 2 x N array, containing letters or gaps

A	-	T	G	G	-	-	A
A	C	-	A	G	A	T	A

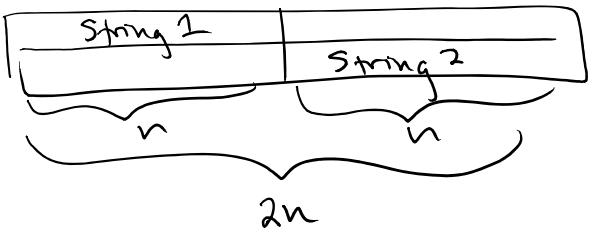
2

N

S. KIMMEL

Q. How large should N be if $n = m$

A) n B) $2n$ C) n^2 D) 2^n



* In optimal solution, put double gaps at beginning